

# Manual de usuario

## AI-Live

Diciembre 2012

## Índice

<b>1</b>	<b>Introducción .....</b>	<b>3</b>
<b>2</b>	<b>Requisitos .....</b>	<b>3</b>
<b>3.</b>	<b>Compilación .....</b>	<b>4</b>
<b>3.1</b>	<b><i>Compilación de toda la aplicación .....</i></b>	<b>4</b>
<b>3.2</b>	<b><i>Compilación por módulos.....</i></b>	<b>4</b>
<b>3.3</b>	<b><i>Instalación y compilación de Ogre3D .....</i></b>	<b>4</b>
<b>4.</b>	<b>Configuración .....</b>	<b>6</b>
<b>4.1</b>	<b><i>Configuración de variables de los módulos .....</i></b>	<b>6</b>
<b>4.2</b>	<b><i>Configuración de Make.....</i></b>	<b>7</b>
<b>5.</b>	<b>Ejecución.....</b>	<b>7</b>
<b>5.1</b>	<b><i>Ejecución manual de clientes IA .....</i></b>	<b>7</b>
<b>5.2</b>	<b><i>Ejecución manual de clientes manuales .....</i></b>	<b>8</b>
<b>5.3</b>	<b><i>Ejecución manual del servidor .....</i></b>	<b>8</b>
<b>5.4</b>	<b><i>Ejecución del cliente GUI .....</i></b>	<b>9</b>
<b>5.4</b>	<b><i>Ejecución utilizando script.....</i></b>	<b>9</b>
<b>5.5</b>	<b><i>Clientes .....</i></b>	<b>10</b>
<b>5.6</b>	<b><i>Servidor .....</i></b>	<b>10</b>
<b>5.7</b>	<b><i>GUI.....</i></b>	<b>10</b>
<b>5.8</b>	<b><i>Ayuda .....</i></b>	<b>10</b>

# 1 Introducción

En este manual se explicarán los requisitos necesarios para ejecutar la aplicación y que funcione correctamente. Posteriormente se detallarán los pasos que se deben realizar para instalar y compilar la aplicación.

También se explicarán las distintas opciones de configuración disponibles para cada módulo de la aplicación.

Por último, se indicará al usuario cómo ejecutar la aplicación, tanto de módulos por separado como por un script que ejecuta todo, y se explicará qué realizar durante la ejecución, según cada módulo.

Este documento está dirigido al usuario final.

# 2 Requisitos

Será necesaria una distribución de Linux. La aplicación ha sido probada desde una distribución **Ubuntu 8.04** de **32 bits**, considerada como la versión más estable ya que la aplicación es compatible con el cliente gráfico 3D. Para versiones posteriores la aplicación funciona pero es posible que existan problemas de compatibilidad con la versión de Ogre. De momento no se ha conseguido realizar la portación de la aplicación en Windows, aunque el código fuente es portable y está preparado para ser ejecutado en otras plataformas, con soporte de socket de BSD.

Si por el contrario se utiliza una versión de Ubuntu de 64 bits, es necesario instalar los siguientes paquetes para conseguir lanzar la aplicación (se puede realizar desde el gestor Synaptic o con “sudo apt-get install”): **ia32-libs** y **ia32libs-dev**.

Será necesario tener instalado **PHP** (versión 5 o superior) con **CLI** (interfaz de línea de comandos), para poder lanzar los ejecutables. No requiere el uso de ningún servidor apache. Se puede descargar desde la web <http://www.php.net/downloads.php> o buscarlo desde el gestor de paquetes Synaptic.

A pesar de que la aplicación contiene archivos de **CLIPS** (.clp), no es necesaria su instalación ya que se encuentra embebido en el lenguaje C. Sí que deberá existir el directorio “../common/clips” que contenga las librerías auxiliares que se usen desde las librerías “../common/clips/clips.h” y “../common/clips/main.h” que se incluyen en el código del cliente.

La aplicación también cuenta con un script de ejecución que se mencionará más adelante, y para su correcto funcionamiento se necesitará tener instalado **xterm** y un **bash** compatible.

Para ejecutar el cliente gráfico 3D se necesita tener instalado **Ogre3D** en su versión **1.6.5**. Posteriormente indicaremos los pasos para realizar la instalación y compilación de Ogre3D.

## 3. Compilación

### 3.1 Compilación de toda la aplicación

Para realizar una compilación del proyecto entero sólo es necesario ejecutar make en el directorio raíz y éste se encargará de realizar el make de todos y cada uno de los módulos del sistema, que incluyen servidor, cliente manual, cliente IA y la interfaz GUI 3D. Para limpiar ejecutables, etc., antes de volver a realizar un make, se ejecuta make clean igualmente en el directorio raíz para que haga el make clean del resto de módulos.

Posteriormente, una vez ejecutado el make hay que comprobar que los ejecutables de los clientes y servidor tienen permisos de ejecución (en el apartado 5 se explica cómo ejecutar cada módulo). En caso de que no los tuvieran, ejecutar chmod para proporcionar permisos de ejecución.

### 3.2 Compilación por módulos

Los módulos disponibles que conforman el proyecto y que tienen un makefile para compilar por separado son:

- Emotional\_engine:** compila el motor de emociones.
- Hability\_engine:** compila el motor de las habilidades.
- Clips:** compila el código clips.
- Server:** compila el servidor.
- Client:** compila el cliente IA.
- Clientmanual:** compila el cliente manual.
- Gui:** compila la interfaz Gui 3D.

Basta con ejecutar make en el directorio en cuestión. También se puede realizar un make clean de cada módulo por separado.

### 3.3 Instalación y compilación de Ogre3D

A continuación explicamos los pasos necesarios para tener instalado y compilado Ogre3D 1.6.

- Acceder desde un terminal al directorio "~/Desktop/": `cd ~/Desktop`

- Instalar los paquetes necesarios:

```
sudo apt-get install pkg-config build-essential autoconf automake libtool libzip-dev  
libxft-dev libxft86vm-dev libxrandr-dev libfreeimage-dev nvidia-cg-toolkit checkinstall  
libfreetype6-dev libpcre3-dev libopenexr-dev freeglut-dev mesacommon-dev libtiff4-dev  
libglademm-2.4-dev libcppunit-dev libxaw7-dev libxaw-headers libois-dev
```

(En caso de que diera error se pueden instalar uno por uno)

- Descargar el plugin CEGUI:

wget <http://downloads.sourceforge.net/crazedsgui/CEGUI-0.6.2.tar.gz>

- Extraer el plugin CEGUI:

```
tar xzf CEGUI-0.6.2.tar.gz
```

- Acceder al directorio "CEGUI-0.6.2/":

```
cd CEGUI-0.6.2
```

- Ejecutar `aclocal`, `bootstrap` y `configure`:

```
aclocal && ./bootstrap && ./configure
```

- Compilar el código e instalarlo:

```
make && sudo make install
```

- Volver al directorio anterior:

```
cd ..
```

- Descargar OGRE3D:

wget <http://downloads.sourceforge.net/ogre/ogre-v1-6-5.tar.bz2>

- Extraer OGRE3D:

```
tar xjf ogre-v1-6-5.tar.bz2
```

- Acceder al directorio "ogre/":

```
cd ogre
```

- Ejecutar `aclocal`, `bootstrap` y `configure`:

```
aclocal && ./bootstrap && ./configure
```

- Compilar el código e instalarlo:

```
make && sudo make install
```

- Volver al directorio anterior:

```
cd ..
```

- Ejecutar `ldconfig`:

```
sudo ldconfig
```

Una vez instalado todo, se pueden borrar las carpetas iniciales del directorio "~/Desktop/".

## 4. Configuración

### 4.1 Configuración de variables de los módulos

La configuración de todos los módulos del proyecto está contenida en el mismo fichero, “confi-ialive”, situado en el directorio raíz de la aplicación. A continuación se detallan las opciones disponibles que se encuentran dentro del fichero:

·Variables globales para todos los clientes y servidor

**rootdir** = directorio raíz del proyecto, contiene los archivos necesarios para compilar y ejecutarlo

**profilesdir** = directorio de los perfiles de personajes

**serverPATH** = directorio donde se encuentra el ejecutable del servidor

**clientPATH** = directorio del cliente

**clientManualPATH** = directorio del cliente manual

**clientGUIPATH** = directorio del cliente gráfico

**ontology** = nombre del fichero de ontología, debe estar en el directorio raíz de la aplicación.

**port** = puerto donde el servidor recibe peticiones

**sizeX** = geometría x de la ventana xterm

**sizeY** = geometría y de la ventana xterm

**posX** = incrementos x para colocar las ventanas xterm

**posY** = incrementos y para colocar las ventanas xterm

**cleanfiles** = nombre del script para la limpieza de archivos

**host** = ip o hostname de la máquina a la que conectar los clientes

·Variables para el servidor

**initial\_state** = initial.state

**server** = ./server

·Variables para el cliente IA

**client** = ejecutable del cliente

**stageC** = escenario por defecto al que conectar

·Variables del cliente manual

**clientmanual** = ejecutable del cliente manual

**stageM** = escenario por defecto al que conectar

·Variables del cliente gráfico

**clientgui** = ejecutable del cliente gráfico

## 4.2 Configuración de Make

Hay un archivo de configuración para make en el mismo directorio raíz del proyecto. Como hemos indicado antes, este make es el encargado de compilar todo el proyecto, que realiza make por cada módulo independientemente. A continuación se detallan sus opciones:

```
export //opción para que todos los make recursivos tengan las variables definidas
CC      = gcc      compilador a usar
PRJCFLAGS = -g      flags generales
RM      = rm orden para borrar archivos
```

Directorios dependientes del directorio raíz

```
eengine_dir = $(CURDIR)/common/emotional_engine
clips_dir = $(CURDIR)/common/clips
server_dir = $(CURDIR)/server-source
client_dir = $(CURDIR)/client-source
clientG_dir = $(CURDIR)/client-gui
clientM_dir = $(CURDIR)/clientmanual-source
common_dir = $(CURDIR)/common
```

```
DIRS = $(eengine_dir) $(clips_dir) $(client_dir) $(server_dir) $(clientM_dir) $(client_gui)
```

En caso de que se añadiera un nuevo módulo, se debería indicar en este fichero el directorio que contiene su propio Makefile con las reglas necesarias para compilarlo.

## 5. Ejecución

Para ejecutar la aplicación, es necesario que se haya lanzado antes el servidor y después los clientes. Hay que abrir una terminal para el servidor y una terminal por cada cliente que se quiera utilizar (tanto de IA, como manual o gráfico). Basta con acceder al directorio de cada módulo y lanzar el ejecutable correspondiente, que comentamos a continuación para cada caso.

### 5.1 Ejecución manual de clientes IA

Para ejecutar un cliente de IA, hay que acceder al directorio “/client-source” (desde el raíz del proyecto) y ejecutar:

```
./client
Usage: client <ontology> <serverhost> <port> [<actor>] [<stage>]
(default stage is DEFAULT_STAGE, default actor is DEFAULT_ACTOR)
```

Los parámetros que se especifican en orden son la ruta a la ontología, ip o nombre del servidor, puerto al que conectarse y como opciones la ruta del perfil del actor a usar (sin la extensión .profile) y el escenario al que conectarse.

Este cliente se ejecuta y realiza acciones automáticamente gracias al módulo de IA, lo que conlleva que no es interactivo con el usuario. Para detener la ejecución del cliente hay que matar el proceso (Ctrl + C). En caso de que se estuviera ejecutando el cliente con un cliente gráfico, se debe cerrar este último primero. Al finalizar la ejecución se creará un archivo de texto con la traza cuyo nombre es el identificador que se asignó al actor. Este archivo contendrá las acciones que ha realizado una por una por el escenario.

## 5.2 Ejecución manual de clientes manuales

Para ejecutar un cliente manual, hay que acceder al directorio “/clientmanual-source” (desde el raíz del proyecto) y ejecutar:

```
./client
```

```
Usage: client <ontology> <serverhost> <port> [<actor>] [<stage>]
```

```
(default stage is DEFAULT_STAGE, default actor is DEFAULT_ACTOR)
```

Los parámetros que se utilizan son los mismos que para un cliente de IA.

Un cliente manual es interactivo, es decir, el usuario es el que decide las acciones que va a realizar el actor. Por cada acción realizada, la aplicación mostrará por pantalla las diferentes acciones (identificadas por un número) que puede realizar, junto con los objetos involucrados en la acción. El usuario debe introducir el número de la acción y pulsar ENTER.

Al contrario que un cliente IA, para finalizar la ejecución no es necesario matar el proceso, ya que una de las acciones que se muestran siempre por pantalla es la de Salir (número 0). Sólo hay que introducir el 0 y pulsar ENTER para salir del programa. En caso de que sólo quedaran ejecutando el cliente manual y un cliente GUI, el GUI debe cerrarse antes.

## 5.3 Ejecución manual del servidor

Para lanzar el servidor hay que acceder al directorio “/server-source” y lanzar el ejecutable:

```
./server Usage: server <ontology> <initial.state> <port>
```

Los parámetros que acompañan al ejecutable son: ruta de la ontología, estado inicial para generar el escenario y el puerto de escucha.

El servidor, al igual que un cliente de IA, se lanza y se ejecuta automáticamente, asignando los turnos a los distintos clientes. Por la terminal se podrá ver la ejecución que va realizando. Para finalizar la ejecución del servidor hay que matar el proceso (Ctrl + C). Se recomienda finalizar las ejecuciones de todos los clientes antes de matar el proceso del servidor.



## 5.4 Ejecución del cliente GUI

Para utilizar el cliente gráfico es necesario haber lanzado antes un servidor y al menos un cliente (de IA o manual). Una vez lanzados se puede ejecutar el gui accediendo al directorio “/client-gui/” y a continuación ejecutar la sentencia “./gui”.

La primera vez que ejecutemos la aplicación después de haber instalado y compilado Ogre3D, y hayamos elegido utilizar un cliente gráfico, nos aparecerá un cuadro para elegir diferentes opciones de configuración de Ogre, como son:

- **Display Frequency**: indica la frecuencia de actualización de la interfaz gráfica.
- **FSAAs (Full-Scene Anti-Aliasing)**: indica el nivel de difuminación de la escena.
- **Full Screen**: se utiliza para elegir si se desea ejecutar la interfaz gráfica en pantalla completa o en ventana.
- **RTT (Render To Texture) Preferred Mode**: indica el tipo de buffer de renderizado a emplear para la representación de la escena.
- **VSync (Vertical Synchronization)**: se utiliza para elegir si se desea ejecutar la sincronización vertical o no.
- **Video Mode**: muestra las opciones de resolución de pantalla (en pixeles) para ejecutar la interfaz gráfica.

Posteriormente si se desea modificar estas opciones hay que ir al directorio “/client-gui” y abrir el fichero “ogre.cfg”, que contiene los valores de estas opciones.

Una vez con el cliente GUI corriendo, se puede alternar entre las distintas perspectivas del escenario o cámaras pulsando las teclas 1-2-3-4 o 5. Con la tecla ESC se cierra la interfaz gráfica (no el juego). Si quedaran ejecutando sólo un cliente (IA o manual), y uno GUI, para finalizar la ejecución hay que cerrar antes el GUI.

## 5.4 Ejecución utilizando script

Para lanzar el programa, tan sólo hay que ejecutar el script run.sh y unos parámetros determinados para lanzar los distintos módulos. Esta es la forma de uso y su explicación.

```
./run.sh -C [server|manual|client] -s -c [profile] -m [profile] -g
```

La opción -C sirve para limpiar trazas y ficheros de estadísticas de los clientes y servidor, tiene tres opciones, server client y manual que lanzan a su vez el script de limpieza de cada uno de esos módulos.

Ej: ./run.sh -C server -C client

Esta orden limpiaría las trazas y ficheros de estadísticas del cliente y los estados que se generan en el servidor.

## 5.5 Clientes

La opción `-c` (profile) lanza un cliente de IA utilizando el perfil que se indique. El archivo del perfil se debe escribir sin la extensión propia `".profile"`. Los perfiles se leen del directorio que se haya especificado en el archivo de configuración. Si se desea saber cómo añadir nuevos campos se debe consultar el apartado "3.Configuración" del Manual de referencia.

Ej: `./run.sh -c DEFAULT_ACTOR`

Este ejemplo lanzaría un cliente de inteligencia artificial usando el perfil `DEFAULT_ACTOR.profile` que se debe encontrar bajo el directorio configurado en el archivo `"conf-ialive"` y con las opciones necesarias para realizar la conexión al servidor y puerto predeterminados.

Para lanzar un cliente manual en el script hay que utilizar la opción `-m` (profile), especificando igualmente el perfil del actor a utilizar.

Ej: `./run.sh -c mike -m DEFAULT_ACTOR`

Este ejemplo ejecuta un cliente de IA utilizando el perfil `mike.profile` y un cliente manual con las características del perfil `DEFAULT_ACTOR.profile`

## 5.6 Servidor

Opción `-s` se utiliza para ejecutar un servidor con las opciones del archivo `"conf-ialive"` Nótese que solo puede haber un servidor escuchando en el mismo puerto de una máquina aunque podría haber varios servidores utilizando diferentes puertos de escucha.

Ej: `./run.sh -s`

Como ya se comentó anteriormente, es necesario que como mínimo se lancen el servidor y un cliente. Por eso la primera opción será la de lanzar el servidor.

## 5.7 GUI

Opción `-g` activa el cliente gráfico 3D con las opciones especificadas en su archivo `gui.ini` que está en el mismo directorio que el binario. Las opciones de ese archivo incluyen la ip o nombre del servidor al que conectarse, puerto y escenario que representará.

Ej: `./run.sh -s -c mike -m DEFAULT_ACTOR -g`

## 5.8 Ayuda

Opción `-h` muestra una pequeña ayuda con los parámetros disponibles. Si se ejecuta el script sin ningún parámetro también se muestra esta ayuda de uso.